Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина «Модели данных и системы управления базами данных»

«К ЗАЩИТЕ ДОПУСТИТЬ»
Руководитель курсового проекта
ассистент
_____А.В. Давыдчик
___.____.2023

# ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту
на тему:

# «ПЛАТФОРМА ПРИВЛЕЧЕНИЯ ФИНАНСИРОВАНИЯ ПРОЕКТОВ»

БГУИР КП 1-40 04 01 016 ПЗ

Выполнил студент группы 053504
МАЗУРЕНКО Данила Александрович

_____
(подпись студента)

Курсовой проект представлен на проверку ___.____.2023

_____
(подпись студента)

Минск 2023

# СОДЕРЖАНИЕ

**потом…**

# ВВЕДЕНИЕ

В современном мире динамичного экономического развития и технологических инноваций, привлечение финансирования для реализации проектов становится ключевой задачей для предпринимателей, стартапов и творческих инициатив. Однако, в условиях сильной конкуренции и изменяющейся финансовой парадигмы, традиционные методы финансирования не всегда способны удовлетворить потребности новаторов и предпринимателей. В связи с этим, появление и рост популярности краудфандинговых платформ приобретает особенное значение.

Краудфандинг – это не только новый способ финансирования проектов, но и мощный инструмент, способствующий активизации сообщества и мобилизации ресурсов, которые ранее могли показаться недоступными. Важной составной частью этой инновационной модели является эффективное управление данными и информацией. Базы данных, играющие важную роль в архитектуре краудфандинговых платформ, обеспечивают функциональность и надежность всей системы.

Цель данной курсовой работы состоит в исследовании платформы привлечения финансирования проектов, с акцентом на роль баз данных в её архитектуре. Будут рассмотрены решения и технологии, которые используются для сбора, хранения, анализа и управления данными на краудфандинговых платформах, и как эти аспекты влияют на эффективность, безопасность и устойчивость таких систем.

Таким образом, данная работа направлена на освещение важной роли баз данных в архитектуре платформы привлечения финансирования проектов и исследование их влияния на успешность и эффективность таких платформ.

# 1 ОБЗОР МОДЕЛЕЙ БАЗ ДАННЫХ

*SQL* (*Structured Query Language*) и *NoSQL* (*Not Only SQL*) представляют собой два различных подхода к управлению данными. *SQL* основан на реляционных базах данных и предлагает строгую структуру и схему для хранения и организации данных. *NoSQL*, с другой стороны, предоставляет более гибкие и распределенные методы хранения и управления данными, и не требует фиксированных схем данных. Далее будут рассмотрены основные различия между этими двумя подходами, а также их преимущества и недостатки в различных контекстах.

База данных SQL – это реляционная база данных который организует данные в таблицы со строками и столбцами. SQL означает язык структурированных запросов, который является стандартным языком, используемым для запроса и управления данными в реляционной базе данных.

Некоторые ключевые характеристики базы данных SQL включают в себя:

– Данные хранятся в таблицах, содержащих строки и столбцы. Каждая строка представляет запись, а каждый столбец представляет атрибут этой записи.

– Между таблицами существуют связи, которые обеспечиваются с помощью внешних ключей. Это обеспечивает целостность данных и уменьшает избыточность.

– Язык SQL используется для запроса данных и управления ими. SQL предоставляет такие команды, как SELECT, INSERT, UPDATE и DELETE, для взаимодействия с базой данных.

– Свойства ACID (атомарность, согласованность, изоляция, долговечность) применяются для обеспечения надежности и целостности данных. Транзакции либо завершаются полностью, либо не завершаются вообще.

База данных NoSQL — это нереляционная база данных который хранит данные в формате, отличном от строк и столбцов. Базы данных NoSQL бывают разных типов в зависимости от модели данных. Основные типы:

– Хранилища «ключ-значение». Данные хранятся в неструктурированном формате с уникальным ключом для извлечения значений. Примеры: Redis и DynamoDB.

– Базы данных документов: данные хранятся в формате документа, например JSON. Примеры: MongoDB и CouchDB.

– Базы данных графов. Данные хранятся в узлах и ребрах, оптимизированных для взаимоотношений между данными. Примеры: Neo4j и JanusGraph.

– Столбчатые базы данных: данные хранятся в столбцах, а не в строках. Примерами являются Cassandra и HBase.

## 1.2 Различия между SQL и NoSQL

Одно из основных отличий между SQL и NoSQL — используемый язык. SQL означает язык структурированных запросов, который с 1970-х годов превратился в мощный язык запросов к структурированным данным. NoSQL — это новая система баз данных, которая не использует стандартный язык запросов, а использует документы JSON для хранения данных. NoSQL предлагает различные модели взаимодействия: от хранилищ «ключ-значение» до баз данных с широкими столбцами, что позволяет использовать различные способы взаимодействия с данными.

**1.2.1** Масштабируемость и производительность

С появлением больших данных потребности баз данных быстро переросли возможности баз данных SQL. В результате была создана технология NoSQL для решения проблем масштабируемости.

Масштабирование базы данных SQL обычно предполагает увеличение вычислительной мощности текущего оборудования, тогда как масштабирование базы данных NoSQL часто предполагает добавление дополнительных серверов или узлов из-за ее первично-вторичной архитектуры.

Базы данных SQL обычно используют горизонтальную масштабируемость, которая включает в себя сегментирование, при котором таблицы делятся на более мелкие разделы и распределяются по нескольким серверам. Amazon Relational Database Service и другие поставщики используют эту популярную форму масштабирования реляционных баз данных.

Базы данных NoSQL используют вертикальную масштабируемость для повышения производительности за счет добавления ресурсов на один сервер. С NoSQL вы можете масштабировать свою базу данных вверх или вниз в зависимости от ваших требований, что дает вам большую гибкость. Вертикальная масштабируемость популярна для облачных приложений, которые помогают более эффективно управлять такими ресурсами, как вычислительная мощность и хранилище. В качестве бонуса этот тип

масштабирования обходится дешевле, чем горизонтальное масштабирование, обеспечиваемое большинством баз данных SQL.

В конечном счете, эффективность используемых структур данных может существенно повлиять на масштабируемость больше, чем различия между базами данных SQL и NoSQL, поэтому крайне важно понимать конкретный вариант использования и соответствующим образом планировать.

**1.2.2** Структурные различия

Базы данных SQL и NoSQL имеют совершенно разные свойства и структуры. База данных SQL — это, по сути, табличный формат, который чем-то похож на электронную таблицу Excel, где каждая строка представляет собой запись в базе данных, а каждый столбец — это поле данных. Связи между полями данных устанавливаются таблицами в базе данных.

Хотя NoSQL может показаться противоположностью SQL, на самом деле это общий термин, который означает «Не только SQL» и относится к базам данных, которые не основаны на табличных отношениях. Базы данных NoSQL активно хранят данные в виде документов, представляющих собой записи, состоящие из наборов ключей или свойств со значениями. Они обладают более гибкой структурой, которая позволяет хранить связанные элементы вместе, не требуя создания таблиц, как это необходимо в базе данных SQL.

Базы данных SQL более строги в использовании предопределенных схема, что ускоряет их использование в транзакционных приложениях. Напротив, базы данных NoSQL не имеют предопределенной схемы. Их можно легко адаптировать к различным типам наборов данных, что делает их идеальными для больших наборов данных и аналитика в реальном времени.

**1.2.3** Свойства базы данных

Каждый тип базы данных имеет свой собственный набор свойств, которые делают его подходящим для определенных случаев использования. Базы данных SQL соответствуют свойствам ACID (атомарность, согласованность, изоляция и долговечность), которые гарантируют точность и надежность обработки транзакций. Напротив, базы данных NoSQL следуют теореме CAP (согласованность, доступность и толерантность разделов), подчеркивая доступность и устойчивость разделов над согласованностью.

**1.2.4** Поддержка и сообщества

Когда дело доходит до поддержки и сообществ, базы данных SQL и NoSQL обладают достаточными ресурсами. Однако из-за своей популярности и того факта, что он существует с 1970-х годов, SQL имеет более широкую поддержку и большее сообщество. Следовательно, найти опытных

специалистов, умеющих работать с SQL, может быть проще, чем найти тех, кто имеет опыт работы с базами данных NoSQL. Кроме того, многие университеты преподают SQL в своей учебной программе по информатике, тогда как NoSQL преподают очень немногие.

С другой стороны, многие базы данных NoSQL имеют открытый исходный код или являются проприетарными и содержат множество ценной документации. Эта обширная документация облегчает разработчикам и инженерам освоение NoSQL. Со временем, когда все больше проектов будут использовать базы данных NoSQL, в отрасли будет расти число опытных специалистов. Крупные технологические компании, такие как MongoDB, предлагают экспертные услуги, а другие компании предоставляют учебные программы для устранения пробелов в знаниях при смене технологий.

**1.2.5** Некоторые популярные типы баз данных SQL:

– Oracle: Запатентованная коммерческая система управления базами данных, широко используемая в корпоративных средах. База данных Oracle предоставляет такие функции, как соответствие требованиям ACID, поддержку SQL и возможность обработки больших объемов данных.

– Microsoft SQL Server: система управления реляционными базами данных, обычно используемая в средах Windows. Microsoft SQL Server предлагает такие функции, как соответствие требованиям ACID, поддержку SQL и интеграцию с другими продуктами Microsoft, такими как Excel и SharePoint.

– PostgreSQL: Мощная система управления реляционными базами данных с открытым исходным кодом, часто используемая для веб-приложений. PostgreSQL предоставляет такие функции, как Соответствие ACID, поддержка SQL и расширяемость с помощью определяемых пользователем функций и хранимых процедур.

– MySQL: система управления реляционными базами данных с открытым исходным кодом, обычно используемая в веб-приложениях. MySQL предлагает такие функции, как соответствие ACID, поддержку SQL и высокую производительность для рабочих нагрузок с большим объемом чтения. Корпорация Oracle теперь владеет MySQL.

**1.2.6** Некоторые популярные типы баз данных NoSQL:

– Хранилища документов: примеры включают MongoDB, Couchbase и Apache CouchDB. Они хранят полуструктурированные или неструктурированные данные в документно-ориентированном формате, где каждый документ содержит набор пар ключ-значение или пар ключ-массив.

– Графические магазины: примеры включают Neo4j, JanusGraph и Amazon Neptune. Они активно используют графовые базы данных для хранения и запроса графовых данных. Элементы данных представлены в виде узлов, ребер и свойств. Отношения между ними исследуются с помощью графовых алгоритмов.

– Хранилища "ключ-значение": примеры включают Redis, Amazon DynamoDB и Riak. Они активно хранят простые данные в формате «ключ-значение», позволяя извлекать значения данных с использованием уникального ключа.

Стоит отметить, что другие типы баз данных NoSQL, такие как хранилища семейств столбцов и объектно-ориентированные хранилища, служат конкретным случаям использования.

### 1.3 Преимущества и недостатки баз данных SQL

Преимущества SQL:

– Хорошо структурированные запросы: Базы данных SQL используют структурированный язык запросов, что делает его идеальным для сложных задач обработки данных.

– Простота в использовании: SQL легко изучить и использовать новичкам.

– Гибкая схема: Базы данных SQL имеют очень гибкую схему, позволяющую управлять различными типами данных.

– Совместимость с популярными языками программирования.: SQL совместим с популярными языками программирования, такими как Java, Python и C#.

Недостатки SQL:

– Ограниченная масштабируемость: Базы данных SQL имеют тенденцию испытывать трудности с горизонтальным масштабированием, а вертикальное масштабирование с использованием более крупных серверов может оказаться дорогостоящим.

– Структурированные данные: Базы данных SQL хорошо работают только со структурированными данными, поэтому, если у вас есть неструктурированные данные или данные, которые часто меняются, управлять ими может быть сложно.

– Ограниченная гибкость: Базы данных SQL имеют фиксированную схему, что затрудняет внесение изменений в структуру данных.

## 1.4 Преимущества и недостатки баз данных NoSQL

Преимущества NoSQL:

– Более простая горизонтальная масштабируемость: Базы данных NoSQL легко масштабируются по горизонтали, что более рентабельно, чем вертикальное масштабирование большого сервера на SQL, что существенно.

– Быстрые обновления и запросы: NoSQL позволяет быстро обновлять или запрашивать большие наборы данных без необходимости перезагрузки всей базы данных.

– Гибкие схемы: Базы данных NoSQL имеют гибкие схемы, что упрощает управление сложными структурами данных.

– Поддерживает неструктурированные данные: Базы данных NoSQL поддерживают различные типы неструктурированных данных, такие как аудио-/видеозаписи и тексты на естественном языке.

Недостатки NoSQL:

– Менее зрелый: По сравнению с базами данных SQL и NoSQL они менее развиты и менее известны.

– Более сложные запросы: Запросы в базах данных NoSQL писать сложнее, чем в базах данных SQL.

– Меньше поддержки транзакций: Транзакции необходимы для обеспечения согласованности данных. Базы данных NoSQL часто не поддерживают их так же, как базы данных SQL.

## 1.5 Вывод

В заключение, сравнивая SQL и NoSQL, можно отметить, что оба подхода имеют свои сильные стороны и применение в различных сценариях. В рамках разрабатываемой краудфандинговой платформы, которая оперирует финансовой информацией и подразумевает строгую структуру данных, SQL базы данных оказываются более подходящим выбором. Реляционная модель SQL обеспечивает высокую надежность и целостность данных, что особенно важно при управлении финансовыми транзакциями и личными данными пользователей. Кроме того, SQL позволяет проводить сложные аналитические запросы, что может быть необходимо для эффективной работы с данными нашей платформы. В совокупности, SQL базы данных способствуют обеспечению безопасности, точности и структурированности данных, что делает их предпочтительным выбором для нашего приложения.

# 2 ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Приложение реализовано на языке JavaScript с использованием технологии React. В качестве среды разработки была выбрана Webstorm IDE. Для создания и работы с базой данных была использована СУБД Postgress.

JavaScript – это высокоуровневый, интерпретируемый язык программирования, который широко используется для разработки веб-приложений. Он обеспечивает интерактивность на веб-страницах, управление браузером, а также может выполняться на сервере (например, с использованием Node.js) для создания бэкенд-части веб-приложений. JavaScript выполняется в браузере клиента или на сервере через интерпретатор, что позволяет динамически изменять поведение веб-приложений без необходимости перекомпиляции. Часто используется для управления веб-страницами, обработки событий, валидации данных, создания анимаций и других клиентских задач. Существует множество библиотек и фреймворков, таких как jQuery, React, Angular и Vue.js, которые упрощают разработку веб-приложений с использованием JavaScript. Кроме того, JavaScript может быть использован для создания серверных приложений с помощью платформы Node.js, что позволяет разработчикам создавать полноценные веб-серверы и взаимодействовать с базами данных. Важными аспектами JavaScript являются асинхронное программирование, синтаксис, системы управления пакетами (например, npm), и активное сообщество разработчиков, которое продолжает развивать этот язык программирования.

React – это библиотека JavaScript, разработанная и поддерживаемая Facebook, которая широко используется для создания интерактивных пользовательских интерфейсов в веб-приложениях. React представляет собой декларативную библиотеку, которая позволяет разработчикам описывать, как должен выглядеть интерфейс в зависимости от состояния приложения, и React берет на себя задачу обновления пользовательского интерфейса при изменении этого состояния. React ориентирован на создание компонентов, которые могут быть переиспользованы. Компоненты представляют собой строительные блоки пользовательского интерфейса и могут быть вложены друг в друга. React использует виртуальный DOM (Virtual DOM), чтобы минимизировать количество манипуляций с реальным DOM, что делает приложение более производительным. Вместо того, чтобы обновлять каждый элемент интерфейса напрямую, React обновляет виртуальное представление и затем сравнивает его с реальным DOM, применяя только необходимые изменения.

Дополнительно к уже сказанному о React, можно упомянуть следующие аспекты:

– Серверный рендеринг: React поддерживает серверный рендеринг (Server-Side Rendering, SSR), что позволяет создавать веб-приложения с быстрым начальным загрузочным временем и улучшенной оптимизацией для поисковых систем.

– Разнообразие сред разработки: Существует множество инструментов для разработки с React, включая Create React App (CRA), Next.js и Gatsby, которые упрощают структурирование проектов и добавляют дополнительные функциональности, такие как роутинг и статическая генерация страниц.

– Мобильная разработка: React Native - это связанный с React фреймворк, который позволяет разработчикам создавать мобильные приложения для iOS и Android, используя знания и компоненты React.

– Расширенные инструменты разработчика: Современные браузеры предоставляют расширенные инструменты разработчика для отладки React-приложений, включая компоненты DevTools для просмотра и отладки структуры и состояния компонентов.

– Современные JavaScript возможности: React активно использует современные возможности языка JavaScript, такие как стрелочные функции, деструктуризация и синтаксический сахар (JSX), что упрощает создание и понимание кода.

– PostgreSQL является мощной, открытой системой управления реляционными базами данных (СУБД). Вот некоторые ключевые аспекты Postgres:

– Открытое и бесплатное ПО: PostgreSQL является свободным и открытым программным обеспечением с лицензией PostgreSQL, что означает, что его можно использовать, изменять и распространять бесплатно.

– Реляционная база данных: PostgreSQL предоставляет мощную реляционную базу данных, поддерживающую SQL (Structured Query Language). Она позволяет создавать и управлять таблицами, связями, индексами и запросами для хранения и извлечения данных.

– Транзакции и целостность данных: PostgreSQL предоставляет надежную поддержку транзакций, что позволяет обеспечивать целостность данных и безопасность в случае сбоев.

– Масштабируемость: Postgres обеспечивает возможность масштабирования с помощью различных методов, включая горизонтальное масштабирование с использованием репликации и разделения данных.

– Расширения и плагины: Сообщество разработчиков создает множество расширений и плагинов для расширения функциональности PostgreSQL под конкретные потребности.

– Поддержка множества языков программирования: PostgreSQL позволяет создавать хранимые процедуры и функции на различных языках программирования, включая SQL, PL/pgSQL, Python, Java, и другие.

– Поддержка ACID: Postgres поддерживает принципы ACID (Atomicity, Consistency, Isolation, Durability), что обеспечивает надежность и стабильность базы данных.

PostgreSQL является популярным выбором для приложений, которые требуют надежного хранения данных, масштабируемости и расширяемости. Он широко используется в различных сферах, включая веб-разработку, приложения для анализа данных, географическую информационную систему (ГИС), и многое другое.

# 3 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА

Разработка краудфандинговой платформы представляет собой теоретически обоснованный и актуальный процесс, обусловленный несколькими ключевыми факторами и принципами:

– Доступ к финансированию: Краудфандинговая платформа предоставляет возможность доступа к финансированию для широкого круга проектов и инициатив, которые могли бы остаться незамеченными или недофинансированными в традиционных финансовых институтах. Этот принцип вытекает из идеи инклюзивности и равного доступа к финансовым ресурсам.

– Развитие инноваций: Краудфандинг платформы способствуют стимулированию инноваций и творчества, предоставляя место, где новаторы и предприниматели могут представить свои идеи и проекты, привлечь финансирование и превратить их в реальность. Это способствует развитию экономики и увеличению конкурентоспособности.

– Разнообразие проектов: Краудфандинг платформы охватывают широкий спектр проектов, включая искусство, технологии, социальные и экологические инициативы, научные исследования и многие другие области. Это способствует разнообразию проектов и идей, способствуя культурному и социальному развитию

– Участие сообщества: Краудфандинговая платформа обеспечивает участие широкой аудитории, позволяя обычным людям стать частью процесса принятия решений и влиять на развитие проектов. Это укрепляет связи между проектами и их поддержкой и способствует социальной активности.

– Эффективное использование ресурсов: Краудфандинговая платформа способствует более эффективному использованию финансовых ресурсов, так как инвесторы могут выбирать проекты, которые соответствуют их интересам и целям, а проекты получают финансирование от тех, кто действительно верит в их успех.

Все эти факторы являются основой для разработки краудфандинговой платформы, которая способствует демократизации финансирования, стимулирует инновации и разнообразие проектов, и укрепляет участие сообщества в процессе решения, какие проекты заслуживают поддержки.

# 4 АРХИТЕКТУРА РАЗРАБАТЫВАЕМОЙ БАЗЫ ДАННЫХ

# ЗАКЛЮЧЕНИЕ

# СПИСОК ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ